Database Optimizing Services

Ghencea, Adrian and Gieger, Immo
Titu Maiorescu University of Bucharest

20. December 2010

Database optimizing services

Adrian GHENCEA, Immo GIEGER
University Titu Maiorescu Bucharest, Romania
Bodenstedt-Wilhelmschule Peine, Deutschland

Almost every organization has at its centre a database. The database provides support for conducting different activities, whether it is production, sales and marketing or internal operations. Every day, a database is accessed for help in strategic decisions. The satisfaction therefore of such needs is entailed with a high quality security and availability.
Those needs can be realised using a DBMS (Database Management System) which is, in fact, software for a database. Technically speaking, it is software which uses a standard method of cataloguing, recovery, and running different data queries. DBMS manages the input data, organizes it, and provides ways of modifying or extracting the data by its users or other programs. Managing the database is an operation that requires periodical updates, optimizing and monitoring.
Keywords: *database, database management system (DBMS), indexing, optimizing, cost for optimized databases.*

1. Introduction

The purpose of the document is to present representative notions about basic optimizing for databases, using mathematical estimation for costs in different types of queries, a review of the level of attained performances, and the effects of different physical access structures in specific query examples. The target group should be familiar with SQL and basic concepts in relational databases. This way, execution strategies for complex queries can be made, allowing the use of knowledge for obtaining information at a lower cost. A database goes through a series of transformations until its final use, starting with *data modelling, database designing and development,* and ending with *its maintenance and optimization.*

2. Database modelling

Data modelling

The data model is more focused on the data that is required and the way in which those should be organised and less on the operations that will be made on the data. The data modelling stages involve the structure, the integrity, the manipulation and the query. There are multiple assets regarding this, such as:

1. Defining the way in which the data should be organised (hierarchical network, relational and object-focused). This provides a definition of rules that restrict what instances of the defined structure are allowed/premises.
2. Offers a data updating protocol.
3. Offers a method for data queries.

A simple structure of data communication which is easily understandable by the final user is the actual result of data modelling.

Customized databases/ Database development

The databases are developed and customized to answer the demands of the customer. The importance of custom databases is major because through them the commercialization of the products of services directly to the target customer becomes possible. The quality of a database is maintained through regular updates.

Database designing

If databases have any of the following problems: malfunction, insecure or inaccurate data or the database has degraded and lost its flexibility, then that is the moment for a new database.

Therefore, the types of specific data and the storage mechanisms have to be defined in order to ensure the integrity of the data through rules and mechanisms of correctly applying the operational principles. All databases should be constructed in regard to the specifications of the customers, including its user interface and functionality. Using the data by including them into a website is possible.

Data mining

Data mining is 'the science of extracting useful information from larger datasets and databases. Every organization wants its business and undergoing processes optimized for the best productivity. The business processes that require optimization include Customer Relationship Management (CRM), Quality Control, Prices and Delivery System etc. Data Mining is a data exploiting discipline which underlines the errors in those processes, using sophisticated algorithms. Data Mining is done on processed data and includes the analysis and determination of the mistakes.

Database Migration

Database Migration represents the transfer (or migration) of basic database schemes and data into the database management, such as Oracle, IBM DB2, MS-SQL Server, MySQL etc. There is a database migration system which allows the reliability and integrity of the data. The migration from one database platform can be difficult and time consuming because of the differences between the standards. However, quick migration of the data between different databases that ensures the integrity of the data is possible, without the loss of any data. Ensuring the access to the data and its protection is essential, especially when large quantities of data or important applications are being moved between systems. The security, availability and reliability of the data can be assured through providing experience in the projecting and applying the database infrastructure with oracle or Microsoft SQL server.

Database Maintenance

Database maintenance is a very important process in every organization. After a secure development of the database, the next process of major importance is the maintenance of the database, which offers an update, a backup and high security.

We could ask ourselves, why does a company need database maintenance?

When a database becomes altered, it is easily observed that the records no longer reflect the reality. This problem usually occurs in case of database deterioration. To remove any doubt regarding the

integrity of the index a manual update and a regular backup is recommended.

As the activity of the organization grows, so does the dimension of the database. A useful practice is to periodically remove unusable data thus increasing the access to the database. Compression of the database will allow easier data supply and simplistic handling of the relevant information from the database.

The same database can be maintained in such way that it will offer the correct result for different questions. For example, the same discussion list can be utilised for extracting the correspondence addresses, the email addresses.

3. Database optimization

Databases are ubiquitous in the modern world. The notion of 'informational library', which is persistent, redundant and well distributed, has become the most important concept in the IT field. As a matter of fact, many people interact with a database management system at a certain level, often without using a computer in every moment of the day.
On each access undergo millions of data transfers, database optimization being a key researching domain for university research institutions, as well as for corporate organizations. From a software development company point of view, the relational databases often serve the software applications in that domain, and the lack of optimization sustains significant costs for both the customer and the company. With millions of data transfers per second, the optimization comes as a surprise and thus represents a key research domain.

The database optimization allows a better configuration and faster searches results. Occasionally the database may present problems such as the failing to provide the requested result, or slow execution. That may make the acquisition of a server necessary. A similar role might have the operating system under which the database cannot be optimized.

The current database infrastructure can be revised thus establishing the best optimization approach and planning for better working environment efficiency.

Through the execution of a database quality control, it can be optimized without duplicates and with high integrity.

Fig. 1. Database schema

Nowadays, this optimization represents a real challenge, especially when the software is constantly changing. However, the database administrators offer relevant solutions to meet their clients' requirements.

Database administration applications
There are different approaches to database administration, and there also are different

ways of optimizing the databases for performance boost, which will also improve the used server. The optimization will depend on the database management system. Each system has its own facilities for optimization. There are programs which have the role of collecting and analysing the required data in order to use the optimization process. These applications will be used in a more alert way as the optimization of the database will become increasingly noticeable. The database systems become more and more important so a continuous database update is mandatory in order to keep up with the changes in the IT domain.

Indexing

One of the ways in which a database can be optimized is indexing. This is made to increase the performance of the queries, which can vary from a database to another, but, generally speaking, all of them benefit from efficient indexes. The efficient indexes allow the queries to avoid scanning the entire structure tables in order to identify the solution. This can be realised with the Microsoft SQL servers.

The SQL server has been made for such sets of indexes. Moreover, its update is permanent, in order to allow the most efficient decisions in query processing. The expert can provide suggestions regarding the way in which the performances of the queries can be increased. The performance of the database must also be updated, so the changes in the dynamic systems have to be taken into account.

A database management system such as Oracle offers its own way of updating. It includes an SQL-type 'adviser' and additionally an access 'adviser'. Those are used to improve the SQL, which is used in package applications. It uses samples in order to collect the necessary data for updates.

The optimization is one of the important ways in which you can keep your systems at optimal performances. They can have different names, but essentially they contribute to the performance increase of the system.

The database optimizers are included in the software which the web holders can use. They represent more complex ways that only IT specialists may use. Nowadays, the applications offer characteristics that increase the optimization's efficiency. In order to be able to maintain the life cycle of the database, the holders have to assure that their databases are advanced.

Using indexes in a database for optimization

A database index is a physical access structure for a database table which works as its name suggests: it is a sorted file that informs the database of the whereabouts of the registrations, which are located on the disc. In order to better understand what an index does, please consider reading a textbook. In order to find a certain section, the reader can read the book until he identifies what he is looking for, or alternatively can check the 'contents' and find the desired section. A database index can work much longer than a textbook index. Adding adequate indexes for large tables is the most important part in optimizing a database. The creation of a unique index for a large table which contains no indexes can reduce the execution time of a query considerably.

*As example, we shall take the following scenario: supposing we have a table of a database named '**EMPLOYEES**' with 100.000 registrations. If we want it to execute the next simple query on this un-indexed table:*

SELECT First Name, Last Name FROM EMPLOYEES WHERE EmpID=12345;

For the purpose of identifying the registration of the employee with the ID aforementioned, the database has to scan the entire 100.000 registrations in order to return the correct result. This way of scanning is usually known as a full-scan of the table. Fortunately, a database developer can create an index on the EmpID column to prevent such scans. Furthermore, in the case of a unique constraint of this domain, the database will compile the physical address of each employee in a table. As such, the scanning becomes pointless, and the localization of the registration is made in real time. After the developer adds this index, the database can locate the registration of the employee with EmpID=12345, which is a potential reduction of 100.000 operations.

Types of indexes

Indexes fall in one of the two categories: clustered or nonclustered. The main difference between the two categories is that the nonclustered indexes do not affect the indexes' ordering located on the hard while the clustered indexes do. Because clustered indexes do affect physically the order of the registrations from the disc, there can be an indexed cluster for each table. The same restriction cannot apply to nonclustered indexes, thus creating space on the disc is possible (though it does not represent the best solution).

Cost estimating for optimized databases

The cost estimating is the process of applying a consistent and significant execution measure of the costs for a certain query. Different metrics can be used for this purpose, but the most relevant and the most common metric is **the number of block accesses query carts.** Since on the disk the inputs/outputs represent a time-consuming operation; therefore the objective is to minimize the number of block accesses, without the sacrifice of functionality.

Estimation of Select operations cost, estimation of Join operations cost, nested Loop, single Loop (using an index) and sort-merge Join can be taken into account.

There are a series of database optimizing methods. Each has ultimately as its result the reduction of the algorithm's complexity. One of the techniques used for that is the Greedy techniques.

The ***greedy*** algorithms are generally simple and are used in optimization issues (for example – finding the easiest path in a graph). In most situations we have:

- Lots of elements (vertices of the graph, works in progress etc.);
- A function that checks whether a mass of candidates is a possible, not necessarily optimal, solution;
- A function that checks whether is possible to complete it for a mass of candidates in order to a possible, not necessarily optimal, solution;

- A selection function that chooses the best unused element at any given time;
- A function that notifies the user that a solution has been reached.

To solve the problem, a *greedy* algorithm builds the solution step-by-step.

The Greedy technique states that the number of configurations (of the nodes and arcs in the graph) is exponential with the number of the candidate structures of the workload; at the same time the main algorithm is not feasible in the case of workloads with a large number of candidate structures.

At the roots of the GREEDY technique of number of configuration reductions stands the GREEDY algorithm:

- If the workload is a sequence then the algorithm is named GREEDY-SQ;
- The GREEDY-SQL algorithm uses a UnionPar function;

The UnionPar function

- Allows 2 solutions p_1= [a_1, S_1,..., a_N, S_N, a_{N+1}] şi p_2= [b_1, S_1,..., b_N, S_N, b_{N+1}]
  For the sequence [S_1,...,S_N];
- Generates a new solution for the same sequence;
- At each K stage, there are generated additional configurations, starting from a_k and b_k configurations which are added in the graph;
- The exit is the shortest path in the generated graph.

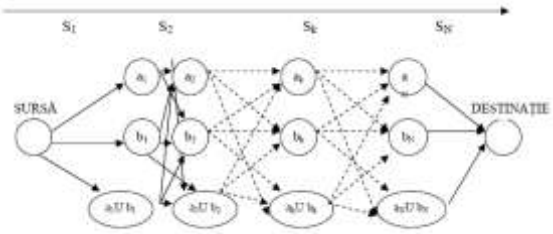

Fig. 2. Graph generated by the function UnioPair

The GREEDY-SEQ algorithm

Step 1. For each structure from S= {s_1, s_2,..., s_M} the optimal solution is found using the main algorithm. There are a lot of P solutions for individual structures.
Let P={p_1,..., p_M} and p_i= [a_{i1}, S_1,..., S_N, a_{iN+1}];

Step 2. Let C be the amount of all the configurations over the individual structures;

Step 3. On the P amount the greedy search is run.

Step 3a. Let r = [c_1, S_1,...,c_N, S_N, c_{N+1}] solution from P where **COST(r)** minimum. P=P-{r}.

Step 3b. We choose **s from P** for which **t=UnionPar(r,s)** has the cost of execution minimal for all elements from P, and **COST(t) < COST(r).**

If s does not exist proceed to **Step 4.**

P=P-{s}, P=PU{t} **goto Step3a.**

Step 4. The graph with all the configurations from P from that stage will be generated, after which the algorithm for the minimum path is run and the solution is given.

4. Conclusions

We use algorithms and techniques which lower the complexity of the centralised databases. This document has as purpose a better perception of the database optimizations for the developer, as well as the way in which a database (ex. DBMS) formulates executional strategies for different types of queries, even though the presented examples are limited in scope. It should also be noted that a well-created database should contain indexes and criteria for the selection of the columns for indexes.

References

[1] I. Lungu and A. Bara, *Executive Information Systems*, ASE Printing House, Bucharest, 2007.

[2] I. Lungu and I. Tanase, *Optimizing queries in relational databases*, Journal Informatica Economica, nr. 1(13)/2000.

[3] T. Marston, *The Relational Data Model, Normalisation and effective Database Design*, 2005.

[4] J.Date. *An introduction to Database Systems*, Addison Wesley, 2004.

[5] http://www.databaseguides.com/

[6] http://www.vinrcorp.com